Task scheduling to extend platform useful life using prognostics.

Jean-Marc Nicod, Léo Obrecht, Christophe Varnier

Task Scheduling to Extend Platform Useful Life using Prognostics*

Jean-Marc Nicod, Léo Obrecht and Christophe Varnier†
Institut FEMTO-ST, CNRS/UFC/ENSMM/UTBM, BESANCON, FRANCE
[Jean-Marc.Nicod/Christophe.Varnier]@femto-st.fr.

June 2013

Abstract

In this paper, we aim at maximizing the useful life of a heterogeneous distributed platform which has to deliver a given production. The machines perform independent tasks and may be configured with different profiles (one nominal mode and several degraded ones). Depending on the profile, a machine reaches a given throughput. At each time the sum of the machine throughputs that are currently running determines the global throughput. Moreover, each machine is supposed to be monitored and a prognostic module gives its remaining useful life depending on both its past and future usage (profile). The objective is to configure the platform so as to reach the demand as long as possible.

We propose to discretize the time into periods and to choose a configuration for each period. We propose an Integer Linear Programming (ILP) model to find such configurations for a fixed time horizon. Due to the number of variables and constraints in the ILP, the largest horizon can be computed for small instances of the problem. For larger ones, we propose polynomial time heuristics to maximize the useful life. Exhaustive simulations show that the heuristic solutions are close to the optimal (5% in average) in the case where the optimal horizon can be computed. For other platforms with a very large number of machines, simulations assess the efficiency of our heuristics. The distance to the theoretical maximal value is about 8% in average.

Keywords – Condition Based Maintenance, Scheduling, Remaining Useful Life, Integer Programming, Optimal Solution, Heuristics

*This work has been supported by a ENSMM funds and has been performed in cooperation with the Labex ACTION program (contract ANR-11-LABX-01-01)

†alphabetic order

1 Introduction

Maintain production equipment in their best operational condition is one major objective of all manufacturers. Since the last decade maintenance budget has growing up to ensure the availability of equipment. Nowadays it is not yet considered as a cost center, but as a profit lever. Maintenance strategies evolve from corrective maintenance policies where breakdowns were a necessary damage to condition based maintenance. In the latter case, equipment are monitored and maintenance action are scheduled to prevent from failures.

Recent development in prognostics and health management allows to imagine new maintenance strategies that take into account the effective and future state of the equipment. The results already proposed by researchers [7] show that it is possible to estimate the Remaining Useful Life (RUL) of a component or an equipment. This could be possible in the framework depicted by [6]. The architecture described in this paper covers all functions of condition based maintenance from the data acquisition with sensors to the decision making. Due to the growing complexity of equipment, it is able to give information, at the minimum level for a control purpose. Then, it is often possible to collect these data. They can be exploited to know what is the system health state. Moreover, with several techniques recently developed one can predict the future state of the system.

Prognostic system are proposed in the literature. [2] present some methods, for instance based on fuzzy neural networks to determine the RUL of machine components. [8] present also for the same goal an approach based on dynamic bayesian networks. As Byington proposed in [1], prognostic methods can be categorized in three groups. First, methods that are based on physical model of the component or the equipment. Then, data-driven methods are based on the past data. Finally, the third group contents methods that use the experience feedback. Whatever the approach, they all try to estimate the RUL. Consequently, this information leads to a decision phase regarding the maintenance policy of the concerned equipment. Decision could have several forms, preventive intervention, spare part provision, rescheduling of production, etc.

In this paper, we address this kind of decision making issue. Considering that equipment is monitored, a prognostic system is able to give the RUL of the equipment. In the context of this paper, we assume that each equipment has a nominal running mode (the most efficient one) and several degraded modes that allow equipment to extend its RUL with a lower efficiency. Then one question is, how can we control the production rate of a set of equipment for a given global production objective to reach the next preventive maintenance window or to produce as long as possible? To achieve our objective, at each time, the decision problem that we are faced to is to find which equipment should be used and in which mode.

One application of this work could be energy production management of a wind farm. Wind turbines suffer from degradations (e.g., gearbox, blade structure) and consequently they have to be periodically maintained. Maintenance operations require costly equipment (e.g., cranes or special trucks) that has to

be hired ([5]). Therefore, for cost and availability reasons we have both to group the maintenance tasks and to postpone them as long as possible while insuring the required service.

The paper is organized as follows: in Section 2 we give a formal definition of the problem. In section 3 we describe an Integer Linear Programming (ILP) formulation that finds a configuration if this one exists for a given horizon. Sub-optimal approaches are presented in Section 4 and simulation results are shown in Section 5 to illustrate their behavior depending on experimental conditions. We conclude the article in section 6.

2 Framework and decision problem

In this section we present the application, resource and degradation models. We also define the objective function of our optimization problem.

2.1 Application, resource framework and execution model

The application that we address in this paper is a collection of independent and identical tasks. The input can be considered as infinite, that is countable or not. The provided result is a given service level that we measure as a throughput, i.e., number of pieces performed or quantity of matter (q) treated per unit of time (ut). The application is supposed to maintain a given service such as the throughput ρ.

The platform $\mathcal{M} = \{M_j \text{ s.t. } 1 \leq j \leq m\}$ consists of m resources (e.g., machines) that perform the inputs as described by the application. Each machine M_j has n running profiles ($p_{i,j}$ s.t. $1 \leq i \leq n$). A machine running with the profile $p_{i,j} = (\rho_{i,j}, RUL_{i,j})$ is able to produce the throughput $\rho_{i,j}$ during at most $RUL_{i,j}\, ut$. It is worth noting that $RUL_{i,j}$ is time before a failure, thus before the halt of M_j. The higher is the throughput of the resource, the shorter is its RUL. So at a given time t, the throughput of the application ρ' is defined as the sum of each throughput $\rho_{i,j}$ of each machine M_j whose profile is $p_{i,j}$ and that is running at that time t. We assume that the overproduction $\rho' - \rho$ is lost. All of the resources of the platform are not supposed to be in use at any time because of their RUL or because the target throughput ρ can be achieved by using only a subset of the available machines within the platform.

2.2 Profile usage model

Let $(\rho max_j, RULmin_j) = (\rho_{1,j}, RUL_{1,j})$ be the largest available throughput and its corresponding RUL. $RULmin_j$ is the smallest useful life for the machine M_j according to the execution model. Moreover, by construction we have $\rho_{i,j} \times RUL_{i,j} < \rho_{i-1,j} \times RUL_{i-1,j}$ for each profile $1 \leq i \leq n$ for each machine M_j because the more the throughput is decreased the less is the efficiency (amount of work) of M_j. This implies that $(\rho min_j, RULmax_j) = (\rho_{n,j}, RUL_{n,j})$.

2.3 Time discretization

First of all, one way to tackle the problem consists in discretizing the time in periods. For each period of length ΔT units of time (ut), a given service ρ has to be satisfied. This approach is not so far from realistic constraint, since one can imagine that one period could be one day or one week in a real case.

2.4 Objective functions

Our goal is to define the running profile $p_{i,j}$ for each resource M_j and for each period of time so as to achieve a given global throughput ρ as long as possible. As ΔT is the smallest considering period of time in which the service has to be satisfied, let $\mathcal{K}$ and $\mathcal{T}$ be respectively the largest number of periods in which the service is satisfied and the corresponding duration. So we have $\mathcal{T} = K \times \Delta T\, ut$ as the longest period of time in which the platform is able to guaranty the target throughput ρ.

2.5 Motivated example

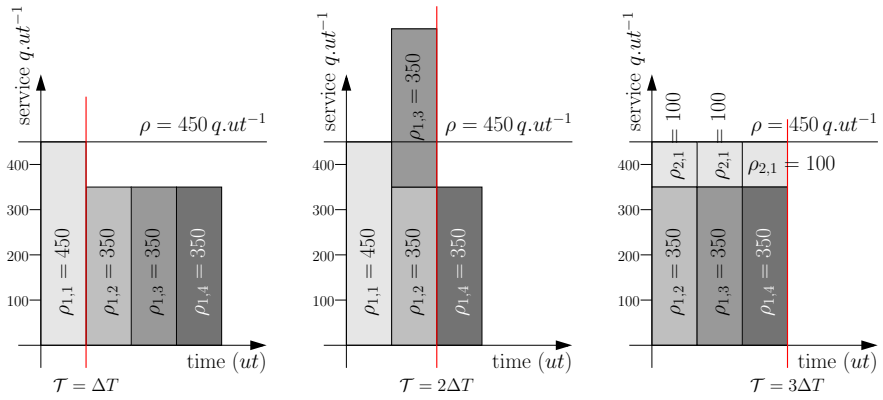
Let us consider a set of four machines ($m = 4$). The required throughput is $\rho = 450\, q.ut^{-1}$. At $t = 0$, M_1 can produce $\rho_{1,1} = 450\, q.ut^{-1}$ with $RUL_{1,1} = 1ut$ or $\rho_{2,1} = 100\, q.ut^{-1}$ with $RUL_{2,1} = 3ut$; M_j ($j = 2, 3, 4$) can produce $\rho_{1,j} = 350\, q.ut^{-1}$ with $RUL_{1,j} = 1ut$ or $\rho_{2,j} = 75\, q.ut^{-1}$ with $RUL_{2,j} = 3ut$. Note that these machines respect the profile usage model. $\forall t > 0$ each profile $p_{i,j}$ ($1 \leq i \leq n$ and $1 \leq j \leq m$) is given in Table 1 for the 3 scenarios presented in Figure 1. This example aims at showing that using a machine within its more efficient profile is not always suitable for improving the useful life of the platform:

- Scenario 1: each machine runs with its more efficient profile without allowing overproduction. Then the platform runs for four periods of time but only one period of time allows us to achieve the target throughput ($\rho = 450\, q.ut^{-1}$). The useful life of the platform is one period ($\mathcal{T} = \Delta T$) for this scenario;

- Scenario 2: each machine is set to its more efficient profile but overproduction is now allowed. The production that exceeds a throughput $\rho = 450\, q.ut^{-1}$ is lost. Nevertheless in this case the production level reaches the target throughput for two periods of time ($\mathcal{T} = 2\Delta T$).

- Scenario 3: we present an optimal scheduling that allows the platform to reach the target demand for three periods ($\mathcal{T} = 3\Delta T$). Machines are not always used in their more efficient profile. Indeed by using M_1 for three periods with a throughput of $100\, q.ut^{-1}$, its production can be joined to the production of one of the other machines with throughput is $350\, q.ut^{-1}$. Because of the small number of alternative scenarios, it is easy to see that any other schedule can not reach the constraint $\rho = 450\, q.ut^{-1}$ for a larger number of periods ($\mathcal{T} \geq 3\Delta T$).

The aim of this example is to show that, when a machine M_j has different running profiles corresponding to different couples $(\rho_{i,j},\ RUL_{i,j})$, it is possible to extend useful life of the platform while respecting a given target throughput. The key point of this problem is to be able to find the appropriate profile for each machine at each period of time.



(a) scenario 1 (b) scenario 2 (c) scenario 3

Figure 1: Motivated example

3 Optimal results

In this section, we propose a first approach of the problem with an exact resolution method, based on an Integer Linear Program (ILP).

3.1 Decision problem

The decision problem we face can be described as follows: does a schedule exist that achieve the given service ρ during a given number of time periods K, considering the current health state of equipment, i.e., the value of $\{RUL_{i,j}$ s.t. $1 \leq i \leq n$ and $1 \leq j \leq m\}$? For this first problem, one can propose an Integer Linear Program ($ILP(\rho, \mathcal{M}, K)$).

3.1.1 Variables

Let $a_{i,j,k}$ s.t. $1 \leq i \leq n, 1 \leq j \leq m$ and $1 \leq k \leq K$ be the decision variables of the problem. $a_{i,j,k}$ is defined as a binary variable. $a_{i,j,k} = 1$ if equipment j is used with the profile i during the period k; $a_{i,j,k} = 0$ otherwise. This set of variables assumes that the profile of the equipment can change at each time period.

Table 1: Profile $(\rho_i, j,\ RUL_{i,j})$ for each machine and each scenario given in Figure 1.

		$t=0$	$t=\Delta T$	$t=2\Delta T$	$t=3\Delta T$
scenario 1 ($\mathcal{T}=\Delta T$)	M_1	(450,1)	(450,0)	(450,0)	(450,0)
		(100,3)	(100,0)	(100,0)	(100,0)
	M_2	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_3	(350,1)	(350,1)	(350,1)	(350,0)
		(75,3)	(75,3)	(75,3)	(75,0)
	M_4	(350,1)	(350,1)	(350,1)	(350,1)
		(75,3)	(75,3)	(75,3)	(75,3)
scenario 2 ($\mathcal{T}=2\Delta T$)	M_1	(450,1)	(450,0)	(450,0)	(450,0)
		(100,3)	(100,0)	(100,0)	(100,0)
	M_2	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_3	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_4	(350,1)	(350,1)	(350,1)	(350,0)
		(75,3)	(75,3)	(75,3)	(75,0)
scenario 3 ($\mathcal{T}=3\Delta T$)	M_1	(450,1)	(450,0)	(450,0)	(450,0)
		(100,3)	(100,2)	(100,1)	(100,0)
	M_2	(350,1)	(350,0)	(350,0)	(350,0)
		(75,3)	(75,0)	(75,0)	(75,0)
	M_3	(350,1)	(350,1)	(350,0)	(350,0)
		(75,3)	(75,3)	(75,0)	(75,0)
	M_4	(350,1)	(350,1)	(350,1)	(350,0)
		(75,3)	(75,3)	(75,3)	(75,0)

3.1.2 Constraints

The constraints of this decision problem should express the production throughput required, the limitation of the useful life and the possible control mode for equipment.

The first set of constraints concerns the production throughput. At least the required service ρ should be reached for each time period. This, can be expressed by the following inequalities:

$$\forall k \quad \sum_{j=1}^{m} \sum_{i=1}^{n} a_{i,j,k} \times \rho_{i,j} \geq \rho \tag{1}$$

The second set of constraints requires that if an equipment M_j is used for a given period k, then it can be controlled using only one running profile $p_{i,j}$:

$$\forall k \forall j \quad \sum_{i=1}^{n} a_{i,j,k} \leq 1 \tag{2}$$

Finally, the last set of constraints is due to the remaining useful life for each equipment. We can consider that during a given period k, if an equipment j is used with the running profile $p_{i,j}$ then, it cuts the remaining useful by $\Delta T / RUL_{i,j}$. Consequently, due to the value of the remaining useful life for equipment j, the following inequalities expressed that equipment could not be

used more than its remaining useful life:

$$\forall j \quad \sum_{i=1}^{n} \frac{\sum_{k=1}^{K} a_{i,j,k} \times \Delta T}{RUL_{i,j}} \leq 1 \tag{3}$$

3.2 Associated optimization problem

The Integer Linear Program $ILP(\rho, \mathcal{M}, K)$ allows without any objective function to give an answer to the question: does a configuration exist for the platform $\mathcal{M}$ such that the required throughput ρ could be reached during at least K periods.

3.2.1 Minimizing the production loss

Nevertheless, one can use this model to obtain solutions with the objective to minimize the loss of production. The loss of production is defined as the difference between the resulting throughput of a given configuration (running mode of equipment) and the required one (equation 4). Then the production loss for a given period k is the following:

$$\sigma_k = \sum_{j=1}^{m} \sum_{i=1}^{n} a_{i,j,k} \times \rho_{i,j} - \rho \tag{4}$$

Consequently, a first optimization problem that we can address is the minimization of the total production loss $\sum_{k=1}^{K} \sigma_k$. This almost corresponds to keep the maximum potential of production for the platform $\mathcal{M}$.

3.2.2 Maximizing the number of periods

As we present in section 2.4, we propose to solve the problem where the platform $\mathcal{M}$ is able to produce the required throughput ρ as long as possible. Besides the previous model can compute a solution to reach the throughput ρ, it is not sufficient since it is designed for a known number of periods K.

Nevertheless it can be useful to determine the greatest number of periods during which a given platform $\mathcal{M}$ is able to produce the given throughput ρ. First, one can determine two bounds of this number. The first one is an upper bound $KMAX$:

$$KMAX = \left\lfloor \frac{\sum_{j=1}^{m} \max_{1 \leq i \leq n} (\rho_{i,j} \times RUL_{i,j})}{\rho} \right\rfloor \tag{5}$$

This equation means that if all equipment are used with their better yield (the running mode that provides the greatest production during the whole remaining useful life) and the global production of the platform is always ρ, then $KMAX$ is the longest duration for which the throughput ρ could be reached.

A lower bound *KMIN* can also be computed using a heuristic algorithm. Then, the worst lower bound is 0. If a heuristic algorithm can provide a solution, the latter could be considered as a lower bound.

Since one can compute these two bounds *KMAX* and *KMIN*, finding the maximum number of periods that can be reached for a given throughput ρ and a given platform $\mathcal{M}$ can be done using a dichotomy search approach. This approach is detailed in algorithm 1.

$ILP(\rho, \mathcal{M}, K)$ is in fact a binary linear program but solving such a binary linear program is a NP-Complete problem. However, as shown in Section 5, efficient solvers such as [4] or [3] are able to give solutions for small problem instances. For more realistic problem size, defining scalable heuristic is mandatory. Thanks to the previous ILP, a validation of these heuristics is presented in the next section.

Algorithm 1: Dichotomy search procedure for finding the maximum number of periods

```
Remark: for this algorithm, we call ILP(ρ, M, K) the integer linear program
described in section 3.1 and LP(ρ, M, K) the rational relaxation
of ILP(ρ, M, K)
K_min ← KMIN
K_max ← KMAX
while K_max − K_min > 0 do
    K ← (K_min + K_max)/2
    if LP(ρ, M, K) has a solution then
        if ILP(ρ, M, K) has at least one solution then
            K_min ← K
        else
            K_max ← K
    else
        K_max ← K
return K
```

4 Sub–optimal approaches

We are not able to compute the optimal solution using the ILP defined in the previous section as soon as we consider platforms within a large number of machines and/or with also a large number of profiles. So we propose four polynomial time heuristics that allocate for each period of time enough machines to reach the target throughput as long as possible. When a machine is chosen, a profile is selected by the heuristic so as to define its contribution to the production within the current period and to compute its remaining useful life when the period is finished. Different approaches are proposed in the following. We distinguish two families of heuristics. In the first family (H–RAND,

H–KS) the schedule is constructed period by period. We update the RUL of the selected machines for the current period and we iterate on this process until the set of available machines that are not able to reach ρ. The second family (H–LRF, H–HTF) consists in finding a selection of machines for several periods corresponding to the smallest RUL of the selected machines. Then each RUL of the selected machines is updated as for the first family. We iterate with the remaining set of available machines.

4.1 H–RAND: Random heuristic

The first heuristic works period by period. H–RAND randomly chooses a machine and its associated profile available for the next ΔT units of time (ut) and iterates as long as the throughput within the current period does not reach at least the demand ρ. The RUL of each machine is updated to take its usage during the current period into account. Then H–RAND selects another subset of machines until a new period can be completed. The number of periods K that are successfully completed represents the useful life of the platform. This naive heuristic mainly serves as a basis for comparaison and assesses the interest of defining more complex and smart heuristics to extend the useful life of the system to a number of periods close to the optimal one.

4.2 H–KS: One period based heuristic

The heuristic H–KS is a more sophisticated heuristic. This heuristic aims at minimizing the production loss, period by period. If we consider one period, the problem is to find a subset of couples machine/profile that is able to reach at least the production demand with the smallest overproduction. We propose to implement a KnapSack-like algorithm so as to make the choice between all the available couples within the current period. The difference with the classical Knapsack problem is first that the sum of the value ($\rho_{i,j}$) of the selected objects (subset of couples machine/profile) should be greater or equal to the knapsack weight (ρ). Secondly each object (M_j) has several values ($\rho_{i,j}$, $1 \leq i \leq n$) and at most one could be selected. The objective of our KnapSack-like problem is now to minimize the sum of the machine values in the case where this sum exceeds the knapsack weight ρ.

The algorithm developed to implement H–KS is a classical dynamic programming based approach. We consider successively each available machine. For each machine M_j we iterate on the throughput ρ' from 1 to ρ. For each value of ρ', we consider each available profile $p_{i,j} = (\rho_{i,j}, RUL_{i,j})$ $(1 \leq i \leq n)$ of M_j to select or not the current machine with its right configuration regarding our objective. To define the objective value we introduce some notations: let $ov_i(\rho', j)$ be the overall throughput obtained by the j first machines using both the j^{th} machine with its i^{th} profile and the optimal configuration considering the $j-1$ first machines obtained for a target throughput of $\rho' - \rho_{i,j}$; let $OV_i(\rho, j)$ be a valide overall throughput and $+\infty$ otherwise; finally let $OV(\rho', j)$ be the optimal (minimal) throughput that exceeds the target demand ρ' using a subset

of the j first machines. The expression of the optimal value is the following:

$$ov_i(\rho', j) = OV(\rho' - \rho_{i,j}, j-1) + \rho_{i,j} \text{ with } 1 \leq i \leq n$$

$$OV_i(\rho', j) = \begin{cases} ov_i(\rho', j) \text{ if } ov_i(\rho', j) \geq \rho' \\ +\infty \text{ otherwise} \end{cases}$$

$$OV(\rho', j) = \min\left(OV(\rho', j-1), \min_{1 \leq i \leq n} OV_i(\rho', j)\right)$$

The minimal throughput for the current period is given at the position $OV(\rho, m)$ of the 2D matrix OV used by our algorithm. Thanks to the storage of each choice that is made for every couple (ρ', j) when the algorithm is running, we are able to reconstruct the way to obtain the optimal schedule.

4.3 H–LRF: Largest *RUL* First heuristic

This heuristic aims at considering each machine M_j using its less efficient profile ($p_{n,j} = (\rho min_j, RULmax_j)$). The idea is to sort the machines by their non-increasing $RULmax_j$ ($\forall 1 \leq j \leq m$), select iteratively the sorted machines by their order until the overall throughput exceeds ρ the less as possible. By construction, the last selected machine (say M_l) has the smallest RUL of the subset of the selected machines because of the sorting. So, this machine determines the number $\lfloor \frac{RUL_{max_l}}{\Delta T} \rfloor$ of periods during which ρ is reached. After these periods of time each profile is updated to take the usage of each machine into account. This heuristic loops on this process. As soon as the remaining available machines are not able to reach ρ, we increase the throughput ($\rho_{n,j}$) of the machines M_j which has the largest RUL from the machines that are yet alive. Its contribution in term of throughput becomes $\rho_{n,j}$ and its remaining useful life $RUL_{n,j}$. So, we iterate this increase in the value of the throughput until the overall throughput reaches ρ. If there are available machines for at least another period, we update their RUL and we start another increasing step as long as possible.

4.4 H–HTF: Highest Throughput First heuristic

The heuristic H–HTF is based on the same principle as H–LRF but now each machine M_j is configured with its more efficient profile $p_{1,j} = (\rho max_j, RULmin_j)$. We sort the machines by an increasing $\rho_{1,j}$ and we select by this order a subset of the smallest number of machines that reaches ρ. Then, as long as possible, we iteratively choose the machine whose RUL is the smallest from the selected machines (say M_l) and decrease its throughput from $\rho_{i,l}$ to $\rho_{i+1,l}$ only if the overall throughput remains greater than ρ. As for H–HTF, the number of completed periods $\lfloor \frac{RUL_{max_l}}{\Delta T} \rfloor$ is given by the selected machine M_l which RUL is the smallest. Then we update the RUL for every machine and we repeat the process until enough machine are able to reach ρ.

5 Simulation results

The both proposed approaches previously described (optimal and heuristic ones) have been validated using random generated benchmarks. First we propose to describe the generation method. Then, when the problem size allows to perform an optimal solution a comparison with the heuristic procedures is depicted. Finally we present results with large size problems.

5.1 Benchmark generation

We consider a set of equipment with the fixed data detailed in table 2. Using this data, we can build each running profile i in function of the number n of profiles desired in the application. The construction model uses the notion of work quantity, which is the product of the remaining useful life by the throughput: $Q_{i,j} = \rho_{i,j} \times RUL_{i,j}$. We define for all equipment j the remaining useful life for each profile i as follows:

$$RUL_{i,j} = \frac{Q_{1,j} - (i-1)\Delta Q}{\rho_{i,j}} \quad 1 \leq i \leq n \text{ and } 1 \leq j \leq m$$

$$\text{where} \quad Q_{1,j} = \rho_{1,j} \times RUL_{1,j} > Q_{n,j} = Q_{1,j} - \rho_{1,j} \times C$$

$$\Delta Q = \frac{Q_{1,j} - Q_{n,j}}{n-1} = \frac{\rho_{1,j} \times C}{n-1}$$

$$\rho_{i,j} = \rho_{1,j} \times \frac{n-i+1}{n} \text{ and } 0 < C < RUL_{1,j}$$

Table 2: Testbed set of equipment

Equipment	$\rho max_j = \rho_{1,j}$	$RULmin_j = RUL_{1,j}$	C
1	300	8	5
2	250	8	5
3	200	10	5
4	175	12	5
5	150	14	5

Then, each problem is generated using the equipment of table 2. One platform is built for a given number of equipment m, a given number of running profile n and a given production throughput ρ to reach during the maximum number of periods. The equipment is randomly chosen in the set proposed in table 2. Consequently, the same equipment could be selected twice or more.

For instance, with $m = 4, n = 3$ and $\rho = 400$ the data set of table 3 could be generated. One can notice that equipment 1 is selected twice and equipment 3 and 4 are not used.

Table 3: Example of data set with $m = 4, n = 3$ and $\rho = 400$

Equipment	running profile	$\rho_{i,j}$	$RUL_{i,j}$
1	0	300	8
	1	200	9.5
	2	100	14
2	0	250	8
	1	166.67	9.5
	2	83.33	14
1	0	300	8
	1	200	9.5
	2	100	14
5	0	150	14
	1	100	18.5
	2	50	32

5.2 Simulation results

The generation protocol used for all the tests proposed hereafter consists of 20 generated instances of problems with values of $n \in \{2, 3\}$, $m \in \{2, 3, 4, 5\}$ and $\rho \in \{400, 410, \ldots, 850\}$. Figure 2 gives the normalization of heuristic solutions with the optimal one computed by the ILP. In this experiment each point represents the mean of 20 instances of the problem with $n = 2$ and $m = 5$. For these results some comments can be done:

- First, one can see that random heuristics seems to be not so bad. One reason is due to the objective function of our problem. All approaches aim at maximizing the remaining useful life of the global platform. Then, at the end of the schedule, the remaining useful life of each machine is almost less than one period.
- H-KS appears to be the best heuristics. Indeed, H-KS, is the only heuristics that computes a local optimal solution for one period. It can be seen that this optimal choice does not deteriorate the overall solution, due to the local objective that minimizes the overproduction.
- Finally, the solutions obtained by H-KS are very good, since they are on average at 5% from the optimal one. And the worst case for this test is about 12%.

An other simulation is shown in Figure 3. It obeys to the same protocol but using a large set of 50 machines considering a throughput between 3 000 and 70 000 and 2 running profiles. Since the optimal solution is not computable using the ILP, this figure represents the normalization with $KMAX$. We recall that $KMAX$ is the theoretical upper bound, larger than the optimal number of periods that is possible to reach in practice. This explains why the results

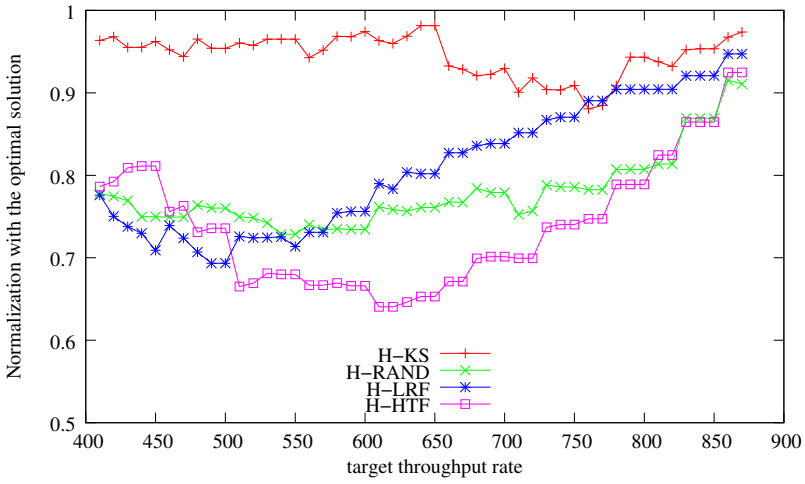


Figure 2: Heuristics H–KS (red), H–RAND (green), H–LRF (blue), H–HTF (purple) ($m = 5$ machines, $n = 2$ running profiles)

shown in this figure are less good as before. Nevertheless, despite the difficulty of the problem, our approach exceeds 80% of *KMAX*.

6 Conclusion and future work

In this paper we investigate a new approach of scheduling tasks on a distributed heterogeneous platform under production throughput constraint. The originality of the approach is that we take machine profiles into account so as to extend the global useful life of the platform. The corollary is that the platform insures the service level during a larger horizon of time. We propose both an ILP formulation of the problem to reach an optimal solution and sub-optimal approaches that are able to provide solutions close to the optimal.

As future work, we plan to explore continuous variation of the machine profile. Moreover, taking maintenance tasks into account within the schedule is also a very challenging problem to solve.

References

[1] C. Byington, M. Roemer, G. Kacprzynski, and T. Galie. Prognostic enhancements to diagnostic systems for improved condition-based maintenances. In *IEEE Aerospace Conf., Big Sky, USA*, 2002.

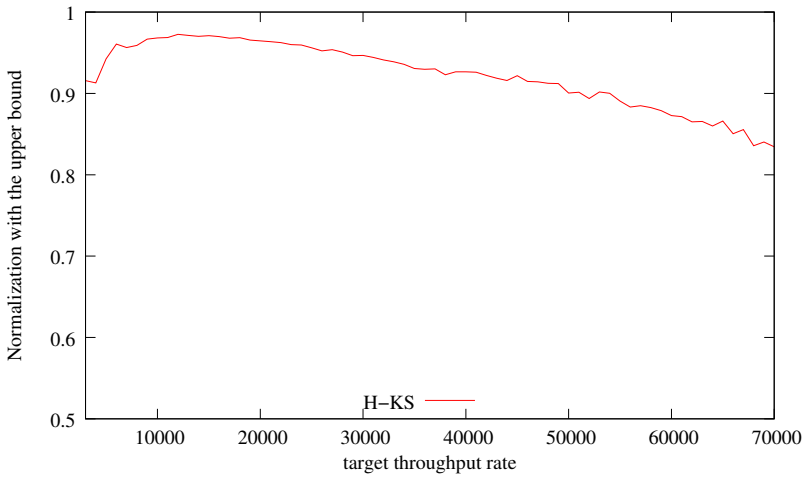


Figure 3: Heuristics H–KS (red)

[2] R. Gouriveau and K. Medjaher. *Maintenance Modelling and Applications*, chapter 2. Prognostics. Part: Industrial Prognostic - An Overview, pages 10–30. Det Norske Veritas (DNV), isbn : 978-82-515-0316-7 edition, 2011.

[3] GUROBI. Optimizer Reference Manual. `http://www.gurobi.com/documentation/5.0/reference-manual/`, 2012.

[4] High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming. `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`, 2012.

[5] A. Kovács, G. Erdös, L. Monostori, and Z.-J. Viharos. Scheduling the Maintenance of Wind Farms for Minimizing Production Loss. In *18th IFAC World Congress*, volume 18, pages 14802–14807, Milano, Italy, sep 2011.

[6] M. Lebold and M. Thurston. Open standards for condition-based maintenance and prognostic systems. In *5th Annual Maintenance and Reliability Conference (MARCON)*, Gatlinburg, USA, 2001.

[7] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao. Intelligent prognostics tools and e-maintenance. *Computers in Industry*, 57:476–489, 2006.

[8] D.A. Tobon-Mejia, K. Medjaher, and N. Zerhouni. Cnc machine tool's wear diagnostic and prognostic by using dynamic bayesian networks. *Mechanical Systems and Signal Processing*, 2012.